Crustal constraint through complete model space screening for diverse geophysical datasets facilitated by emulation

A.W. Roberts[1,2], R.W. Hobbs[1], M. Goldstein[2],
M. Moorkamp[3], M. Jegen[3], B. Heincke[3]

[1]*Dept. of Earth Sciences, Durham University, Durham, DH1 3LE, UK;*
[2] *Dept. of Mathematics, Durham University, Durham, DH1 3LE, UK;*
[3]*IFM-Geomar, Wischoffstrasse, Kiel, Germany*

Abstract

Deep crustal constraint is often carried out using deterministic inverse methods, sometimes using seismic refraction, gravity and electromagnetic datasets in a complementary or "joint" scheme. With increasingly powerful parallel computer systems it is now possible to apply joint inversion schemes to derive an optimum model from diverse input data. These methods are highly effective where the uncertainty in the system is small. However, given the complex nature of these schemes it is often difficult to discern the uniqueness of the output model given the noise in the data, and the application of necessary regularization and weighting in the inversion process means that the extent of user prejudice pertaining to the final result may be unclear. We can rigorously address the subject of uncertainty using standard statistical tools but these methods also become less feasible if the prior model space is large or the forward simulations are computationally expensive. We present a simple Monte Carlo scheme to screen model space in a fully joint fashion, in which we replace the forward simulation with a fast and uncertainty-calibrated math-

ematical function, or emulator. This emulator is used as a proxy to run the very large number of models necessary to fully explore the plausible model space. We develop the method using a simple synthetic dataset then demonstrate its use on a joint data set comprising first-arrival seismic refraction, MT and scalar gravity data over a diapiric salt body. This study demonstrates both the value of a forward Monte Carlo approach (as distinct from a search-based or conventional inverse approach) in incorporating all kinds of uncertainty in the modelling process, exploring the entire model space, and shows the potential value of applying emulator technology throughout geophysics. Though the target here is relatively shallow, the methodology can be readily extended to address the whole crust.
Keywords: Bayesian, statistical methods, emulation, joint inversion, salt diapir, crustal imaging

---

1. Introduction

1.1. Methodological background

A widely used approach for determining deep crustal structure is to use a deterministic non-linear inverse method (Zelt and Barton, 1998; Hole et al., 2006; Roberts et al., 2009). A forward simulator code is used to compute synthetic data and by seeking to minimise an objective function, which normally includes residuals with respect to an observed dataset and some regularization and smoothing terms, an update to the model is computed. This is repeated iteratively until an acceptable value of data misfit, normally measured by the χ^2 parameter, is obtained. Where several kinds of data have been recorded at the same location (e.g. seismic refraction, seismic reflection,

gravity, electromagnetic) these may all be used together to constrain a common model across various physical parameters; seismic velocity, density, resistivity, where the parameters are coupled by some relationship (Moorkamp et al., 2011). The deterministic inverse approach works well when there is clear justification for the use of a particular regularization and smoothing regime and a good prior understanding about the region of model space of where the optimum model is to be found (i.e. the user can be confident that the process will find the global minimum in the objective function, rather than merely a local minimum). In a joint setting the method also works well where there is no uncertainty regarding the parameter coupling. However, often such clarity and certainty is not possible, but in order to obtain a result, overly subjective assertions are made about the degree of regularization, smoothing, coupling, and data uncertainty. In such cases, there is the strong possibility that unquantified user bias may influence the result. So while linearised inversion methods are best suited to obtaining a single optimum result, they do not facilitate a rigorous treatment of the uncertainty associated with a system. In general it is the case that rather than a single optimum model, because of various kinds of uncertainties associated with the system, many structures could give rise to the observed data. Understanding not just an optimum model, but the whole plausible model set, along with a clear understanding of the prior beliefs we are imposing on the constraint process, is important for making inference about the deep crust.

In recent decades, with computational development, and the importance of more fully treating the uncertainty associated with geophysical results, statistical determination schemes are increasingly being used to constrain earth

structure. The most widely known and used methodology is that of the Markov Chain Monte Carlo (MCMC) method, where a point in model space is chosen and parameters updated according to some sampling scheme, such as that of Metropolis-Hastings-Gibbs (Hastings, 1970; Smith and Roberts, 1993), in order to maximise the *likelihood* function. However, although recent computational advances have made it possible to handle larger problems, it is still the case that these methods are still really only practicable when the number of model parameters and the size of the model space is *comparatively* small (Sambridge and Mosegaard, 2002). This is because if the number of parameters is large (commonly $> 10^5$), the number of (complex and therefore time-expensive) forward simulations required to sample the model space and properly build up the posterior probability density functions becomes infeasible, although parallel computing methods are beginning to mitigate this to an extent.

Prior to the development of MCMC methods, authors such as Press (1970) developed conceptually straightforward Monte Carlo methods based on the simple sampling of model space. However, these methods were quickly sidelined in favor of the more targeted sampling strategies of MCMC-related methods, on account of computational efficiency. In this study, we re-adopt the conceptually straightforward approach, of seeking to sample the entire model space, but instead of using the full forward simulator codes, we use *emulators* to rapidly sample the model space and screen it for plausibility. This differs significantly from the search-based MCMC or deterministic inversion methods in that rather than trying to "build up" the plausible model space or search for an optimum model, here the aim is to start with the

entire prior model space and to exclude implausible regions. The result is therefore guaranteed to include *all* potentially plausible model space, given all uncertainties specified as being relevant to the problem at hand.

1.2. What is an emulator?

An emulator is a fast statistical representation of a forward modelling code. By training the emulator with a number of runs of the full simulator code, the emulator seeks to predict the output of the full simulator to a calibrated uncertainty, using a number of simple (in our case polynomial) functions. Emulators are widely used e.g. climate modelling (Rougier et al., 2009), ocean modelling (Logemann et al. , 2004), and cosmological applications (Vernon and Goldstein, 2009). In many cases, their use is crucial to the ability to effectively model the system concerned since the systems are so complex that to run a full simulation of a system such as the earth's atmosphere, ocean, or the universe, would be infeasible on the grounds of insufficient computational capacity, even with recent technological advances. These emulator-based methods have provided considerable insight into the systems concerned through making the modelling process tractable. In this study we use the emulator to screen model space in order to discern all regions containing models representing earth structures which could have given rise to the observed data, given the specified uncertainties in data measurements, physical relationships and any others which the user may wish to specify.

In a number of ways an emulator is similar to a neural network, in that through a process of learning the relationship between model parameters and the data outputs, it seeks to give a rapid prediction of the output of a complex code for a given set of input model parameters. Neural networks have

been used in order to solve inverse problems in geophysics; for example Meier et al. (2007) use a neural network to invert shear wave data. However, an emulator differs from a conventional neural network in that the emulator is fully uncertainty-calibrated. Not only does the emulator give a rapid estimate of the complex forward code output, but it will also give a calibrated estimate of the uncertainty associated with that estimate. This uncertainty calibration makes it possible to use the emulator to test and accept/reject model space for plausibility. Simply having a prediction of the forward code output, without an uncertainty estimate would not allow robust screening of model space, simply because there would be no measure as to the reliability of the output estimates being tested. However, with a calibrated emulator, this problem is resolved. As with a conventional neural network, an emulator will typically run several orders of magnitude faster than the full simulator code, and so may be used to test and accept or reject large areas of model space very quickly. Other authors have also developed methods to quickly approximate the output of a full forward code in order to accelerate inverse methods; James and Ritzwoller (1999) use truncated perturbation expansions to approximate Rayleigh-Wave Eigenfrequencies and Eigenfunctions; and Shapiro and Ritzwoller (2002), who take a similar methodology to use in a MCMC scheme to construct a global shear-velocity model of the crust and upper mantle. In each of these cases the aim is to minimise some objective function or maximise a likelihood function.

Having built an emulator, we generate sets of model parameters using a space-filling sampling design (latin hypercube), and test the emulator estimates of the forward code output for the candidate model parameter-sets

against an observed dataset, in order to reject implausible regions of model space and thus constrain the region of plausible model space. After populating the plausible space, a new emulator can be built over this smaller region, with smaller predictive uncertainty. This new emulator can then be used to further reject implausible model space. We repeat this cyclically until no further emulator uncertainty reduction achieved, at which point all structure has been discerned in the system. An advantage of this kind of approach, which relies entirely on forward modelling, is that it is conceptually straightforward to include any kind of uncertainty or prior belief about the model space. This may include data uncertainty, inter-parameter relationship uncertainty, model discrepancy (uncertainty due to the fact that no model perfectly represents nature). The user simply generates models from a prior space with the required properties.

We present a synthetic example using a 4-layer 1D model space over seismic velocity, density, and resistivity (16 parameters in total, including layer thickness). The resistivity, density and velocity parameters are linked by an uncertain physical relationship (based on Gardner et al. (1974); Jegen-Kulcsar et al. (2009)). Using a lab PC over a period of $\sim$72 hours, after an initial investment of 1000 model runs using the forward *simulator* code, we use 11 emulation cycles to screen $\frac{1}{4}$ billion models over our large starting model space, and reduce it by a factor of 10^{-19}. We then apply the method successfully to an industrial 3D dataset at selected locations.

While the test case here is rather simple, and indeed it would be straightforward to solve using a deterministic inversion method, or indeed sampling the model space using the full simulator, our aim here is to show the value

of a top-down and joint approach, where the entire possible model space is considered, and to present emulation as a potentially useful tool both for facilitating this strategy and more widely in the field of geophysical constraint. Based on our experience and that of Meier et al. (2007) and others, we posit that emulation techniques have the potential to be used widely in the field of geophysics and more generally throughout the Earth Sciences.

2. Methods

The problem being presented is that of joint constraint of a synthetic 1D earth structure. We begin by briefly introducing the three different modelling techniques/domains being used for the study; seismic refraction, magnetotelluric (MT) and gravity and the function of the respective forward simulators, before considering the prior model space and coupling relationship which we will sample in order to constrain the structure from the synthetic dataset.

2.1. Seismic refraction technique

The seismic modelling technique takes advantage of the fact that differing rock types possess differing velocities at which sound waves travel through them. By measuring the time which sound waves take to travel along different paths in the earth, one can gain information about this seismic velocity field, and thus make interpretative judgements about the geological structure in that region.

A full treatment of the seismic method is given in Kennett (2001). In this study, we are considering energy which turns in the earth in a manner described by Snell's law (Equation 1 and Figure 1), due to the presence of a

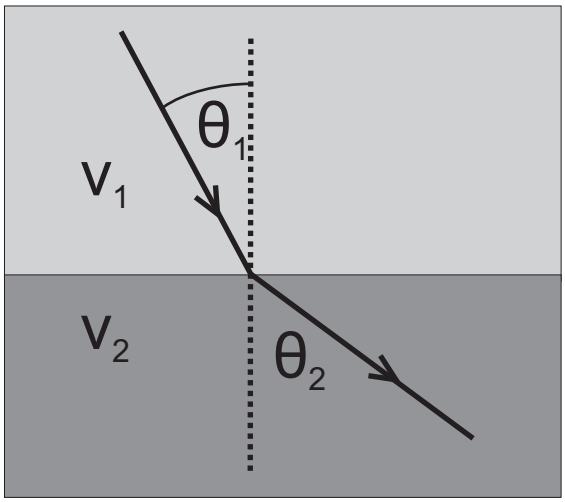

Figure 1: When sound waves encounter an impedance discontinuity, their path is altered. Where the density is unchanged, the deflection is controlled by the velocity, as shown, according to Snell's law (Equation 1). In the limiting case of a velocity gradient (infinitessimally thin layers of increasing velocity), the energy turns in the earth as shown in Figure 2.

velocity gradient, as in Figure 2. For our purposes, we are only considering energy which is NOT reflecting off some boundary/step change in impedance, ie. we are only considering energy which is travelling along paths akin to that of the blue ray in Figure 2. Note that in the scenario being described here, where we are using constant velocity model parameters, in order to create turning waves with the forward simulator, we add a small velocity gradient of 0.5%.

$$v_1 \sin(\theta_2) = v_1 \sin(\theta_2) \tag{1}$$

The forward simulator, which takes the input model and generates a set of output travel time data, is written by Björn Heincke (Heincke et al., 2006)

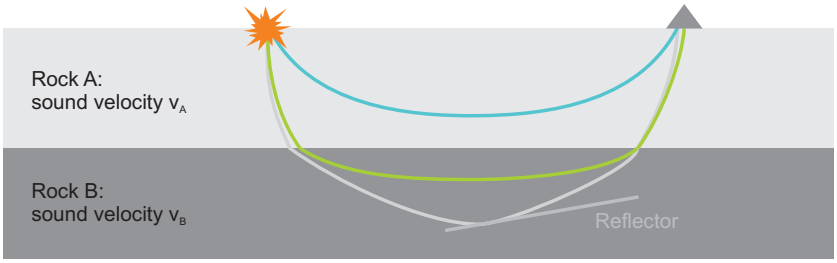

Figure 2: Rays showing the propagation path of sound waves through the earth. Sound waves turn within a velocity gradient (eg. blue and green rays), and they can also reflect from impedance contrasts (grey ray). Here we are considering only seismic energy turning as shown by the blue and green rays, and are not considering reflected energy. Note that in general sound waves have a frequency spectrum with finite width. Describing the process in terms of rays is only strictly valid when considering an infinite frequency spectrum.

and is based on the Finite Element (FE) method described by Podvin and Lecomte (1991). One of the major advantages of using a FE method such as this is that, unlike with a ray-tracing approach, travel times are calculated for ALL receiver locations, rather than simply where the infinite frequency ray-tracing approach finds a ray-path. For our purposes, this means that problems associated with a potentially differing number of data points with each run of the simulator are avoided. For the 1D purpose at hand, this simulator may be considered rather more complex than necessary, and if the aim of this paper were to simply find an efficient solution to the 1D problem, a less complex and computationally intensive code would easily suffice. Indeed, as is commented elsewhere in this paper, an emulator is not strictly necessary to solve this class of problem. However, although we are using a simple earth model, our intention is to show as far as possible an emulator being built for a complex 3D simulator which may be used on a large dataset. The

work was also carried out contemporaneously with that of Moorkamp et al. (2011) with a view to exploring complementarity between the deterministic and stochastic methods. This complementarity is not the main focus of this paper, however in the context of the collaborative nature of the work, we decided to use a common set of simulator codes.

The FE engine requires a gridded volume, and we chose to set this up with dimensions appropriate for a reasonably high resolution large 2D refraction seismic survey. The simulator was thus configured to use a 50,000 x 1,000 x 30,000 m volume with a 100 m grid spacing. Travel times were generated/'recorded' every 500 m along the main axis of the model in a line colinear with the shot position (0 m, 500 m, 10 m). Note that the receiver and shot positions are slightly below the surface of the model, since placing these positions along model boundaries/nodes can cause simulator instability. Because in our example we use 1D layer-parameterized models, the model-input function of the simulator was modified so that it could read in a series of 4 velocity values and 4 layer thicknesses and use these values to generate the required for the simulator. With this parameterization 1,000 runs of the simulator typically took about 4.5 minutes. The output data for each run of the simulator are thus a set of 100 (offset, time) points.

The output data from the simulator for a given model input are a series of travel times recorded at a number of offsets (in our case 100) from the receiver position. Note that this may seem a high density of traveltime points given the problem under consideration. However, in the context of demonstrating the use of the data reduction technique described in Section 2.7.1, and in order to present a data density closer to that seen in 3D datasets, we have

chosen to use a high traveltime density.

2.2. MT technique

The Magneto-Telluric (MT) method aims to probe the resistivity structure of the Earth by measuring the terrestrial electric and magnetic fields at the Earth's surface. Cagniard (1953) give a comprehensive description of the method. The key output parameter from an MT experiment is a measurement of the Earth's complex impedance $Z = \frac{E_x}{H_y}$ over a range of electromagnetic frequencies. The raw output is normally in the form of $R(\omega) = \mathrm{Re}(Z(\omega))$ and $I(\omega) = \mathrm{Im}(Z(\omega))$, however this is normally plotted in terms of the transformed functions *apparent resistivity* r and *phase* ϕ, as defined in Equation 3.

$$\log_{10}(r_{app}(\omega)) = \log_{10}\{(R(\omega)^2 + I(\omega)^2)/(2\pi\mu_0 10^{\omega})\} \quad (2)$$

$$\phi(\omega) = \frac{180}{\pi}\arctan(I(\omega)/R(\omega)) \quad (3)$$

Because the form of the $R(\omega)$ and $I(\omega)$ plots are much simpler in form than the apparent resistivity/phase plots, we choose to consider R and I, as shown in Figure 4.

Our simulator, written by Avdeev et al. (2002), takes as inputs a list of resistivity r and layer thickness s values, along with a list of frequencies, and outputs a list of R and I values, evaluated at each of the values of ω. In our case, we have four layers (so, as with the seismic case, 8 model parameters), and we choose to evaluate the output functions $R(\omega)$ and $I(\omega)$ at 20 frequencies ω_{1-20}. ω is in some sense a proxy for depth in the Earth, in that the MT signal at higher frequency gives information about the upper

structure and the signal at low frequency gives information about the deeper structure. We therefore choose a fairly wide frequency range, from $\log_{10}(\omega) = -6$ to $\log_{10}(\omega) = 0.5$. Because of this large frequency range, we emulate $R(\log(\omega))$ and $I(\log(\omega))$ as a function of the model parameters r_{1-4} and s_{1-4}.

2.3. Gravity technique

This technique uses the fact that the fine-scale gravitational field is sensitive to density variations in the Earth. The gravitational field at a given point, a distance r from a point source of mass m is given by Equation 4.

$$g = -\nabla\phi = -\frac{Gm}{r^2} \tag{4}$$

As with other inverse square law phenomena, we can apply Gauss' theorem (Equation 5) to obtain, in the 1D case, for our four-layer model, with $\vec{\theta} = [\rho_{1-4}, s_{1-4}]^T$, Equation 6.

$$\iint \vec{g} \cdot d\vec{A} = -4\pi G \iiint \rho dV \tag{5}$$

$$g = \sum_{i=1}^{4} \left(\frac{\rho_i s_i}{\left(\sum_{j=1}^{i} s_j\right)^2} \right) \tag{6}$$

Our gravity simulator is extracted from the Full Tensor Gravity (FTG) inversion code by Moorkamp et al. (2011). In this 1D scalar setting, the simulator runs quickly. The output dataset, for a given $\vec{\theta}$ in the 1D case, thus consists of a single point measurement.

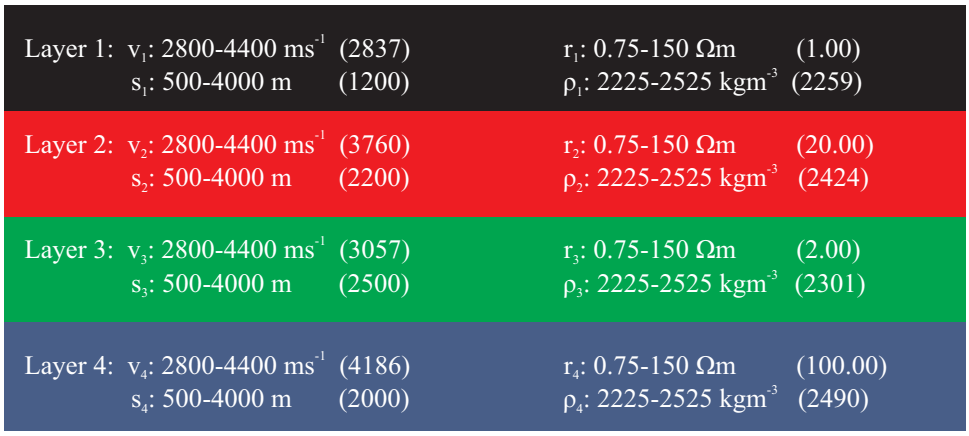

Figure 3: Schematic diagram of the joint model, with parameter ranges and the values used to produce the target dataset given in parentheses.

2.4. Model Space

Before building an emulator to screen model space, the bounds of the model space over which the emulator is to be used, must first be specified. In defining the prior model space, it should be borne in mind that in order to build a reliable emulator, the emulator should be trained over a slightly larger space than is required for screening purposes. This is in order to ensure that the edges of the prior plausible parameter space are sufficiently sampled and that information from these regions is included in the emulator construction. The prior plausible parameter space used here is shown in Figure 3.

2.5. The datasets

The synthetic scenario we are considering is that we have seismic refraction traveltime data, complex impedance data from a MT survey, and a gravity measurement. The seismic and MT datasets, generated using the model parameter set shown in Figure 3, are shown in Figure 4.

In the case of the seismic data, the (x,t) curve is seen to consist of four "segments", reflecting the fact that the synthetic model is parameterized by four constant velocity layers. The discontinuities in gradient contain structural information on account of the fact that deeper-propagating seismic energy emerges at larger offsets. In a layered system, therefore, the positions of these gradient discontinuities can be exploited to constrain the layer thicknesses. We exploit this in Section 2.7.2.

The MT dataset consists of two curves showing $\mathrm{Re}(Z(\omega))$ and $\mathrm{Im}(Z(\omega))$ varying with the driving frequency of the MT instrument. The electromagnetic skin depth is inversely proportional to $\sqrt{\omega}$, and so lower frequency signals propagate further into the Earth.

The gravity datum consists of a single synthetic measurement computed to be 78.73896 mgal. Note that the absolute value of this synthetic measurement should not be interpreted physically because the datum has not been specified. It has simply been computed using the densities and layer thicknesses shown in Figure 3. In a real earth scenario, the measurement includes considerable contribution from the deeper earth, and so for the purposes of earth inference it is the value relative to a datum which is significant. In our case we will be generating candidate density models and computing gravity measurements to test against this synthetic value without any contribution from a half space below, so we are in effect using a zero reference datum. In the real data example presented later, we consider the gravity measurement over the salt relative to that over sediment (Section 3).

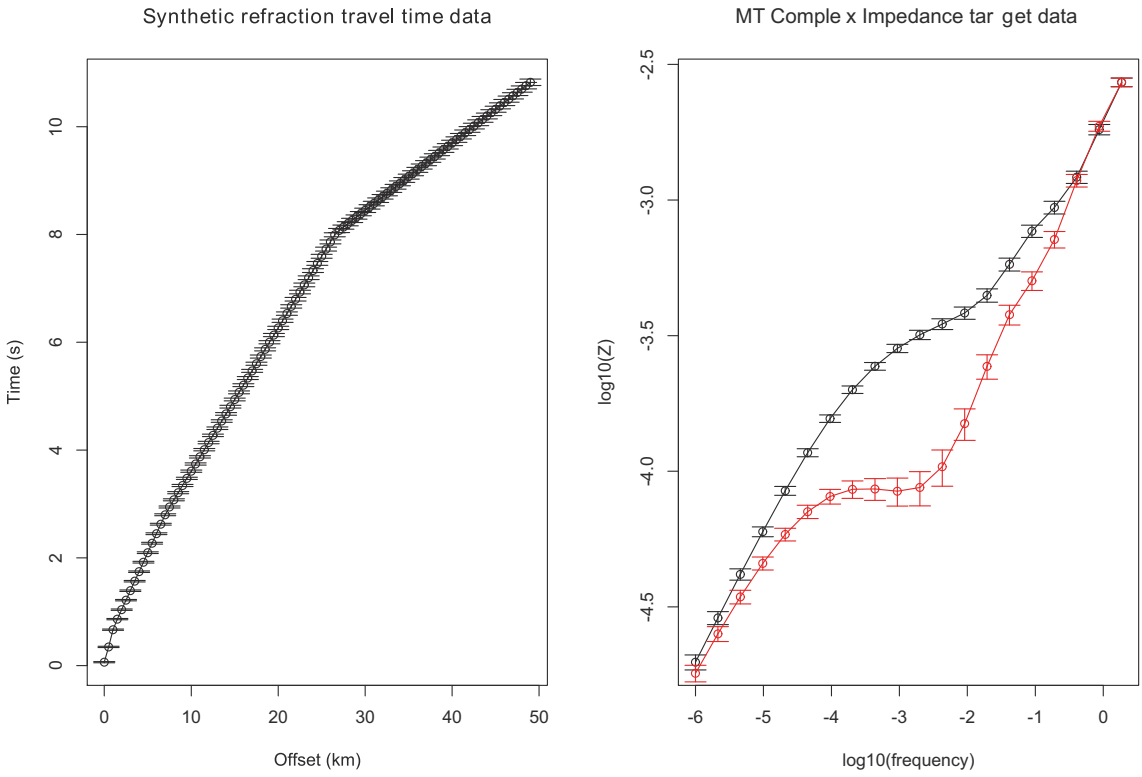

Figure 4: Synthetic datasets used for the study. Left: Refraction travel time data. Note that the travel time curve consists of four distinct straight segments. This is due to the model used to generate it consists of four constant velocity layers. Right: MT Complex Impedance data. The black line/points show $\mathrm{Re}(Z(\omega))$ and the red line/points show $\mathrm{Im}(Z(\omega))$. The data were generated by running the respective forward simulator codes with the parenthesized model parameters shown in Figure 3.

2.6. Relationship uncertainty

At the heart of a joint constraint method is the coupling between the different kinds of model parameters. With the screening approach presented here, any kind of coupling can easily be implemented. Here we build three emulators; one for each of the seismic, gravity, and MT forward simulators. We choose to build each emulator independently of each of the other modelling domains so we do not need to invoke inter-parameter coupling in order to build each emulator. Then, having built an emulator for each modelling domain, we invoke the inter-parameter relationship with a specifed uncertainty to generate joint candidate models for screening, and the emulators are used to discern which of these joint models are commonly plausible in all three domains. It is not strictly necessary to specify such a physical relationship or coupling scheme. However, in such a situation, the "joint" nature of the problem reduces to three independent modelling scenarios. Here we seek to jointly constrain density, resistivity and seismic velocity parameters for each of four layers (Figure 5).

In practice, the relationships linking these parameters are normally derived empirically, by fitting a relationship to pre-existing borehole log data, for example. This means that the relationship between the parameters is uncertain. In linearised inversion schemes, using an uncertain physical relationship can create significant conceptual and technical challenges, however because this method relies entirely on forward modelling, we can naturally include this uncertain relationship by using it to generate our distribution of candidate models at the screening stage.

The coupling relationship shown in Figure 5 applies in a sub-basalt set-

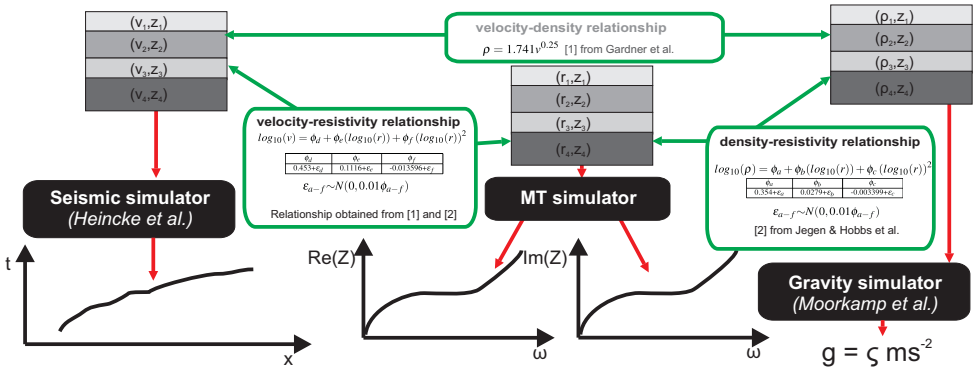

Figure 5: Joint modelling setting.

ting, such as that characteristic of the North East Atlantic in the vicinity of the Faroe Islands. We chose this scenario as a test case because of the particular value a joint approach can add to constraint of this kind of structure. The challenges associated with sub-basalt seismic imaging are well documented (Roberts et al., 2009, for example). Recent technological and methodological advances (Lunnon et al., 2003; Ziolkowski et al., 2001) have given rise to some improvements to intra- and sub-basalt seismic images. However, the typically highly heterogeneous nature of basalt, which gives rise to significant scattering of the seismic wavefield, means that the fundamental challenge remains. Jegen-Kulcsar et al. (2009), among others, have demonstrated the value of joint inversion methods, by virtue of the complementary information provided by different kinds of data.

2.7. Building the emulators

Jegen-Kulcsar et al. (2009) showed the value of using MT and seismic datasets together to constrain a sub-basalt problem. Because of the normally

non-unique nature of seismic models in a sub-basalt context, a Bayesian approach such as this is particularly suited to be applied in this regime. We design each of our emulators in three stages. We first employ a data-reduction method. This is because, for example, in the seismic dataset, which consists of (offset, traveltime) data, the data points, particularly at large offsets are highly correlated. Having reduced the dataset, we then use a least-squares fitting routine in order to fit the output data from a series of training runs of the full simulator code to the model parameters in order to build a predictor. Simply having a predictor, however, is not very useful in practice, however, unless one has an estimate of the uncertainty of the predicted output. We therefore, thirdly, calibrate the uncertainty of the predictor by calibrating it against the output from the full simulator code for the training runs. The result (the emulator) is a framework with which the output of the full forward code can be predicted rapidly, with a calibrated uncertainty estimate. This can then be used to screen candidate model parameter sets for plausibility.

2.7.1. Building a seismic emulator

To construct an emulator, a batch of training runs using the full forward simulator code is required. The seismic simulator code used to train the emulator was that used by Heincke et al. (2006). We generate 1,000 sets of the 8 velocity model parameters (4 velocities and 4 layer thicknesses for each model) using a space-filling latin hypercube design over the model space shown in Figure 3. Each of these 1,000 models is then passed through the seismic *simulator* code, which generates a set of 100 (offset, traveltime) points for each set of model parameters. Because of the typically highly correlated nature of the (x, t) points, we then chose to re-represent each of these travel

time data curves as a set of $p_{seis} = 8$ polynomial coefficients $\alpha_{1-p_{seis},seis}$. The value of p_{seis} was determined by trial and error to obtain the optimum order of polynomial wiith which to fit the (x,t) curves. These coefficients, $\alpha_{i,seis}$, are generated by least-squares fitting each curve to the functional form shown in Equation 7. $g_{seis}(x)$ is an uncertainty function, which we calibrate later.

$$\log(t^2) = \left(\sum_{i=0}^{p_{seis}} \alpha_{i,seis} \left(\log(x^2+1) \right)^i \right) + g_{seis}(x) \tag{7}$$

$$\vec{\theta_{seis}} = \left[\begin{array}{cccccccc} v_1 & v_2 & v_3 & v_4 & s_1 & s_2 & s_3 & s_4 \end{array} \right]^T \tag{8}$$

$$\alpha_{i,seis} = \left(\sum_{k=1}^{w_{seis}} \sum_{j=0}^{q_{seis}} \beta_{ijk} \theta_{k,seis}^j \right) + g_{i,seis}(\vec{\theta_{seis}}) \tag{9}$$

Having re-represented the training datasets as sets of $\alpha_{1-8,seis}$, we seek to predict these coefficients for a given set of model parameters. We do this by fitting the α_i to functions of the model parameters $\theta_{seis,1-8} = (v_{1-4}, s_{1-4})$, using coefficients β_{ijk}, as shown in Equations 8-9. The functions $g_{i,seis}(\vec{\theta_{seis}})$ are uncertainty functions associated with the prediction of each $\alpha_{i,seis}$ coefficient, w_{seis} is the number of model parameters being considered (8), and q_{seis} is the number of β_{ijk} coefficients used to fit each α_i coefficient (4). The value of q_{seis} was determined by trial and error, by examining the optimum order of polynomial for reconstructing the $\alpha_{i,seis}$. Here we have chosen q_{seis} to be the same for each α_i, however this need not be the case in that it would be straightforward to use a different number of β coefficients for each i.

In the coefficients β_{ijk}, we now have a means to predict the output of the simulator code for a given set of input model parameters θ_{seis}. Examples of the ability to predict or reconstruct the data using these β_{ijk} coefficients are shown in Figure 6.

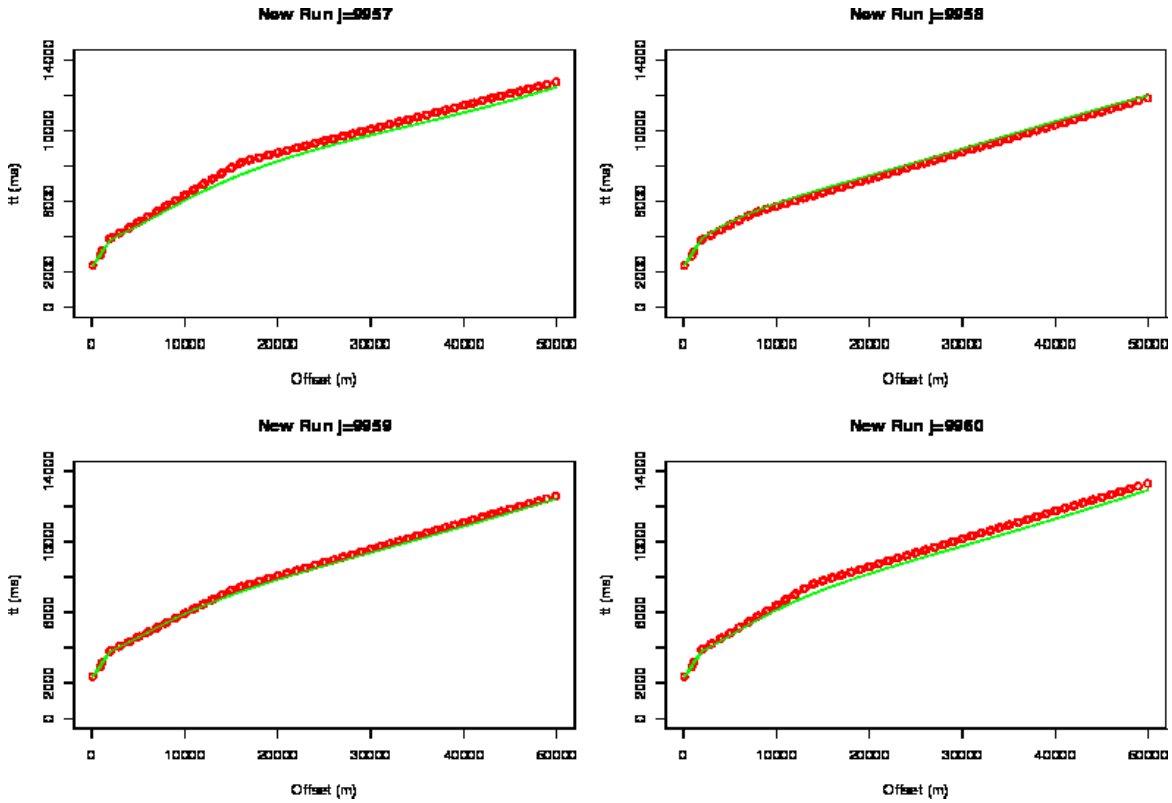

Figure 6: Sample emulator output traveltime data plotted with the real travel times. Red dots mark the real data points and the green lines represent the result of predicting the travel times using the emulator with the relevant model parameters. Note that there is a generally good reconstruction and much of the misfit which is present occurs where there is a discontinuity in the travel time vs offset gradient function (in this case due to the fact that the models consist of four constant velocity layers).

However, simply having a prediction of the output of the forward code is of little value, unless one can also specify the uncertainty of the prediction. Our next step is therefore to calibrate the uncertainty of the predictor output. We do this by using our predictor to generate outputs for the training models and comparing these to the outputs obtained from the full simulator code through construction of a residual function, as shown in Figure 7. Note that the residual function we are constructing here is not simply either $g_{seis}(x)$ or $g_{i,seis}(\vec{\theta_{seis}})$, as in Equations 7 and 9. Rather, it is given by $G_x(x)$ (Equation 14 where n is the number of emulator training runs), which we use to approximate $G(x, \theta_{seis})$ (Equations 12-13). Note that although in this example the traveltime plots themselves consist of four distinct segments (Figure 6) due to the use of constant velocity four layer models, the residual plots do not manifest these segments since over the set of 1000 models used to construct the residuals, different layer thicknesses are used, drawn from across the model space shown in Figure 3.

$$\log(t^2) = \sum_{i=0}^{p_{seis}} \left(\left(\sum_{k=1}^{w_{seis}} \sum_{j=0}^{q_{seis}} \beta_{ijk} \theta_{k,seis}^{j} \right) + g_{i,seis}(\vec{\theta_{seis}}) \left(\log(x^2+1) \right)^i \right) + g_{seis}(x) \tag{10}$$

$$= \sum_{i=0}^{p_{seis}} \sum_{k=1}^{w_{seis}} \sum_{j=0}^{q_{seis}} \beta_{ijk} \theta_{k,seis}^{j} + \left[\sum_{i=0}^{p_{seis}} \left(g_{i,seis}(\vec{\theta_{seis}}) \left(\log(x^2+1) \right)^i \right) + g_{seis}(x) \right] \tag{11}$$

$$= \sum_{i=0}^{p_{seis}} \sum_{k=1}^{w_{seis}} \sum_{j=0}^{q_{seis}} \beta_{ijk} \theta_{k,seis}^{j} + G(x, \vec{\theta_{seis}}) \tag{12}$$

$$\approx \sum_{i=0}^{p_{seis}} \sum_{k=1}^{w_{seis}} \sum_{j=0}^{q_{seis}} \beta_{ijk} \theta_{k,seis}^{j} + G_x(x) \tag{13}$$

$$G_x(x) = \sqrt{\frac{\sum_{n=1}^{n_{max}} \left(t_{em,n}(x) - t_{sim,n}(x) \right)^2}{n_{max}}} \tag{14}$$

2.7.2. The seismic second derivative

It can be seen from Figure 6 that the largest discrepancy between the emulated output t vs x function occurs where there is a discontinuity in $\frac{dt}{dx}$. In our scenario the positions of these gradient discontinuities contain useful information, since the offset can be thought of as a proxy for the depth in the model. As a result, the offset positions of these gradient discontinuities contain information about the depths of the boundaries in our structure.

Rather than seek to emulate the gradient function to probe this information, we calculate the second derivative of the t vs x function (Figure 8), and given we are using 4-layer models, we aim to estimate the offset positions x of the three largest spikes in this $\psi = \left(\frac{d^2t}{dx^2}\right)^2$ function. Note that we use $\left(\frac{d^2t}{dx^2}\right)^2$ rather than $\frac{d^2t}{dx^2}$ in order that the ψ is positive definite, simplifying

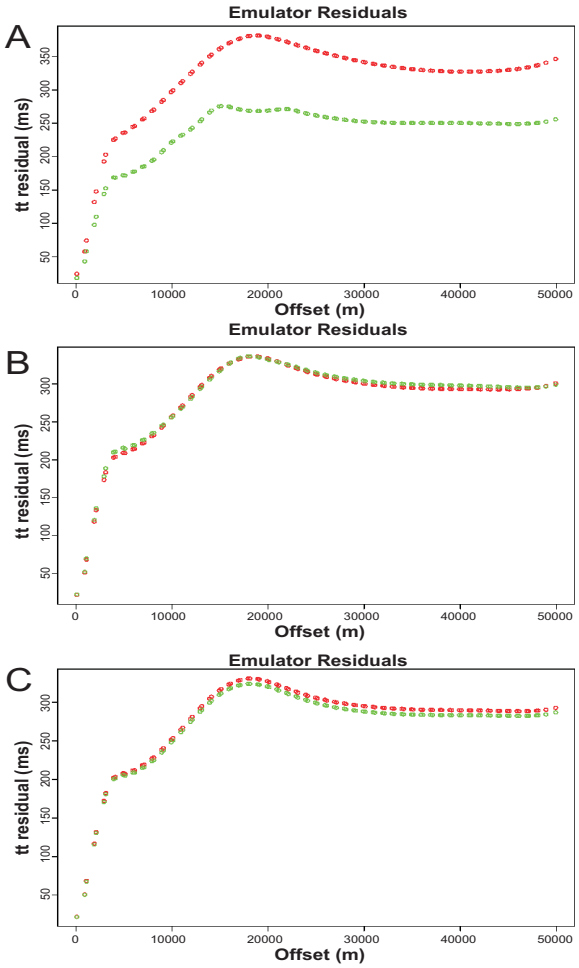

Figure 7: Sample seismic emulator output traveltime data residuals using (A) 100, (B) 1,000 and (C) 10,000 simulator runs to generate the emulator. Green dots are the residuals for the models used to create the emulator and red dots the residuals from running the emulator for 10,000 further models not used for training. Notice two effects: 1) The emulator traveltime residual for new models (red dots) decreases with increased training runs. 2) On using 100 runs, the emulator traveltime residual computed using the models used to construct the emulator is smaller than that for running the 10,000 further models, whereas on using 1,000 training runs, the emulator traveltime residual function for those models well represents that obtained on running further models, implying that ∼1,000 training models are sufficient to cover the model space.

the process of picking the extrema.

We choose to do this because, given we are using a polynomial to represent the t vs x function, if we try to fit a polynomial to the derivative of this function, $\frac{dt}{dx}$, the result of the Least Squares fit is likely to simply be the derivative of the function given by our α-coefficient polynomial representation, which we could calculate analytically, and so we would not gain further useful information. Also, the parts of the gradient function containing the most useful information are the steepest-turning regions, which are the most difficult parts to fit using smooth functions. Another advantage of the 'spike'-fitting approach over trying to predict the gradient function itself is that the number of data points we are aiming to fit for an n-layered model is n-1, so in our case we are trying to fit only three datapoints (the x-positions of the three largest spikes in the $\psi = \left(\frac{d^2t}{dx^2}\right)^2$ function). Having only three datapoints to fit thus makes the emulator construction process considerably more efficient.

2.8. MT and Gravity emulators

A similar method was used to build emulators for the MT and gravity modelling scenarios. The forward simulator codes used for training these emulators were based on those by Avdeev et al. (2002) and Moorkamp et al. (2011), respectively. In the case of MT, the forward simulator generates a complex impedance as a function of frequency $Z(\omega) = \mathrm{Re}(Z(\omega)) + \mathrm{i.Im}(Z(\omega))$, in an analogous way to the seismic simulator, which generates travel times as a function of offset $t(x)$. We therefore use a similar strategy as for the seismic case in order to predict the MT simulator output for a given set of model parameters. Firstly, we reduce the dataset by fitting a set of

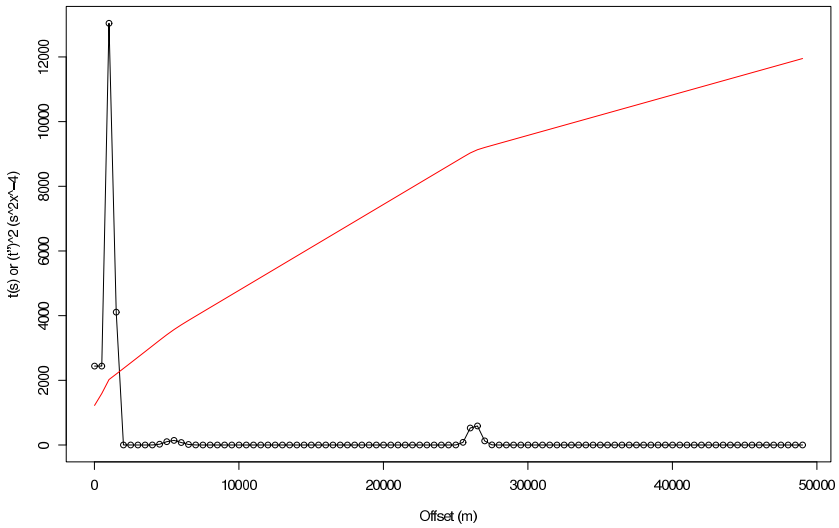

Figure 8: Synthetic t vs x plot (red), with (scaled) $\psi = \left(\frac{d^2t}{dx^2}\right)^2$ overlaid (black). The aim is, to within a known uncertainty, predict the positions of the three largest maxima of the (x,ψ) function. In this example, ψ_{max} are seen at $x \approx 1500$, $x \approx 5500$, $x \approx 27000$.

polynomial coefficients α_{MTR} and α_{MTI} to the simulator outputs $\mathrm{Re}(Z(\omega))$ and $\mathrm{Im}(Z(\omega))$ respectively. We then fit these α_{MTR} and α_{MTI} to the model parameters $\vec{\theta_{MT}}$. The formulation is shown in Equations 15-28.

Emulating $R(\omega) = \mathrm{Re}(Z(\omega))$:

$$\log(R(\omega)) = \left(\sum_{i=0}^{p_{MT}} \alpha_{iR} \left(\log(\omega) \right)^i \right) + g_{p_{MT}R}(\omega) \tag{15}$$

$$\theta_{MT} = \begin{pmatrix} r_1 & r_2 & r_3 & r_4 & s_1 & s_2 & s_3 & s_4 \end{pmatrix}^T \tag{16}$$

$$\alpha_{iR} = \left(\sum_{k=1}^{w_{MT}} \sum_{j=0}^{q_{MT}} \beta_{ijkR} \theta_{ijk,MT}^j \right) + g_{iR}(\vec{\theta_{MT}}) \tag{17}$$

$$\log(R(\omega)) = \left(\sum_{i=0}^{p_{MT}} \left(\left(\sum_{k=1}^{w_{MT}} \sum_{j=0}^{q_{MT}} \beta_{ijkR} \theta_{ijk,MT}^j \right) + g_{iR}(\theta_{MT}) \right) \left(\log(\omega) \right)^i \right) + g_{p_{MT}R}(\omega) \tag{18}$$

$$= \sum_{i=0}^{p_{MT}} \sum_{k=1}^{w_{MT}} \sum_{j=0}^{q_{MT}} \beta_{ijkR} \theta_{ijk,MT}^j \left(\log(\omega) \right)^i + G(\omega, \vec{\theta_{MT}}) \tag{19}$$

$$\approx \sum_{i=0}^{p_{MT}} \sum_{k=1}^{w_{MT}} \sum_{j=0}^{q_{MT}} \beta_{ijkR} \theta_{ijk,MT}^j \left(\log(\omega) \right)^i + G_{\omega,R}(\omega) \tag{20}$$

$$G_{\omega,R}(\omega) = \sqrt{\frac{\sum_{n=1}^{n_{max}} \left(R_{em,n}(\omega) - R_{sim,n}(\omega) \right)^2}{n_{max}}} \tag{21}$$

and $I(\omega) = \mathrm{Im}(Z(\omega))$:

$$\log(I(\omega)) = \left(\sum_{i=0}^{p} \alpha_{iI} \left(\log(\omega) \right)^i \right) + g_{pI}(\omega) \tag{22}$$

$$\theta_{MT} = \begin{pmatrix} r_1 & r_2 & r_3 & r_4 & s_1 & s_2 & s_3 & s_4 \end{pmatrix}^T \tag{23}$$

$$\alpha_{iI} = \left(\sum_{k=1}^{w_{MT}} \sum_{j=0}^{q_{MT}} \beta_{ijkI} \theta_{ijk,MT}^{j} \right) + g_{iI}(\theta_{MT}) \tag{24}$$

$$\log(I(\omega)) = \left(\sum_{i=0}^{p_{MT}} \left(\left(\sum_{k=1}^{w_{MT}} \sum_{j=0}^{q_{MT}} \beta_{ijkI} \theta_{ijk,MT}^{j} \right) + g_{iI}(\theta_{MT}) \right) \left(\log(\omega) \right)^i \right) + g_{p_{MT}I}(\omega) \tag{25}$$

$$= \sum_{i=0}^{p_{MT}} \sum_{k=1}^{w_{MT}} \sum_{j=0}^{q_{MT}} \beta_{ijkI} \theta_{ijk,MT}^{j} \left(\log(\omega) \right)^i + G(\omega, \vec{\theta_{MT}}) \tag{26}$$

$$\approx \sum_{i=0}^{p_{MT}} \sum_{k=1}^{w_{MT}} \sum_{j=0}^{q_{MT}} \beta_{ijkI} \theta_{ijk,MT}^{j} \left(\log(\omega) \right)^i + G_{\omega,I}(\omega) \tag{27}$$

$$G_{\omega,I}(\omega) = \sqrt{\frac{\sum_{n=1}^{n_{max}} \left(I_{em,n}(\omega) - I_{sim,n}(\omega) \right)^2}{n_{max}}} \tag{28}$$

After testing we chose $p_{MT} = 9$ and $q_{MT} = 3$.

Examples of the uncertainty function calibrated over 1000 models after a single emulation cycle are shown in Figure 9.

In the case of gravity, because we are in a 1D setting, there is only a single gravity point. There is therefore no need to perform the first stage, of reducing the dataset, since there is only one point. In the case of gravity, we therefore simply fit the simulator gravity outputs ϕ directly to the model parameters $\vec{\theta}_{grav}$ and calibrate the predictive uncertainty, as in Equations 30-31.

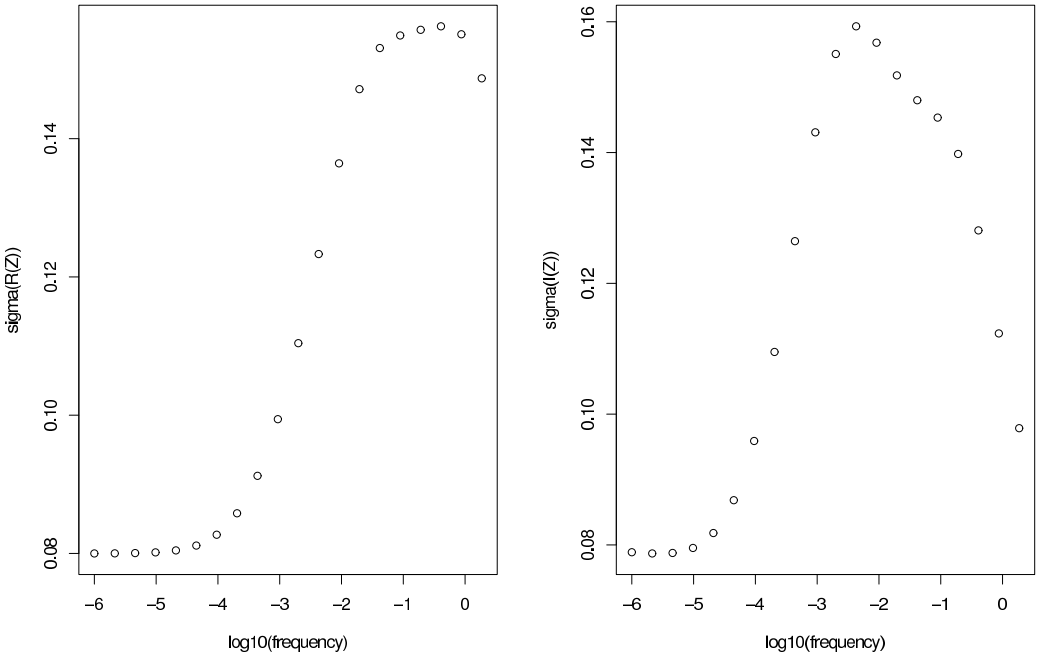

Figure 9: Standard deviation of MT emulator residual at each frequency point for Re(Z) (left) and Im(Z) (right).

$$\phi = \left(\sum_{k=1}^{w_{grav}} \sum_{j=0}^{q_{grav}} \beta_{jk,grav} \theta_{k,grav}^{j} \right) + g_{grav}(\vec{\theta_{grav}}) \tag{29}$$

$$g_{grav}(\vec{\theta_{grav}}) \approx \sqrt{\frac{\sum_{i=1}^{n} \left(g_{i,em} - g_{i,sim}\right)^2}{n}} \tag{30}$$

$$\tag{31}$$

2.9. Using the emulators

Having built emulators for each of the seismic, gravity and MT simulators, we then use the four of them in order to screen model space for implausibility.

Figure 10 shows the results of using a preliminary seismic emulator to screen a set of 10,000 models drawn from the model space over which the emulator was trained (Figure 3). The screening is carried out by generating a candidate model, computing the emulator output predicted dataset, and then comparing the predicted dataset to the target dataset. In the seismic case, for example, if the target dataset lies within $\gamma_{seis} G_x$ of the emulator predicted output, where γ_{seis} is a scaling factor designed to ensure a high probability of the true model parameter set being selected in a synthetic test, then the model parameter set is deemed to be plausible. Here we choose γ_{seis} to be 3, and so the plausibility condition is as shown in Equation 32.

$$\frac{\sum_n \min\left[|(t_{em}(x_n) - t_{targ}(x_n)| - \gamma_{seis} G_x(x_n), 0\right]}{n} < 1 \tag{32}$$

Statistics pertaining to the construction of the emulator are shown in Table 1. Note that using the emulator we can rapidly reduce the plausible model space to 213/10,000 (2.13%) of the original "prior" model space, and that we have done this in a time of around 5 minutes (most of which is used in

building the emulator itself rather than actually screening the models) rather than about 45 minutes, which is how long it would have taken to run the same number of models through the full simulator on the same computer. In order to verify that the system was behaving in a sensible manner, i.e. that it was selecting plausible models for which the full simulator output (as opposed to simply the emulator output), the full simulator data outputs were then computed and plotted for each of the 10,000 models. Figure 10 shows the full simulator outputs for the models which the emulator deemed to be plausible (green) and those it deemed implausible (blue), as well as the target simulated dataset (red). Note that all the simulated data outputs for models deemed plausible by the emulator lie close to the target dataset whereas the simulated outputs for the models deemed implausible by the emulator lie far from the target dataset. This suggests that the emulator is indeed useful for screening the model space, as intended.

The seismic, spike, MT and gravity emulators can together be used to screen model space. Candidate models are generated from the prior model space shown in Figure 3, where the density, resistivity and seismic velocity parameters are linked by the uncertain physical relationship as shown in Figure 5. We then use each of the four emulators to screen these models to discern which models are jointly plausible given our assertions regarding the data uncertainty, relationship uncertainty, model discrepancy and potentially other uncertainties. In this example, because we are using three methods to screen models, we use a starting model pool of 100,000 models. Figure 11 shows the contribution each method is making to constraining the plausible space. If the user wishes to bias the selection weighting towards one of

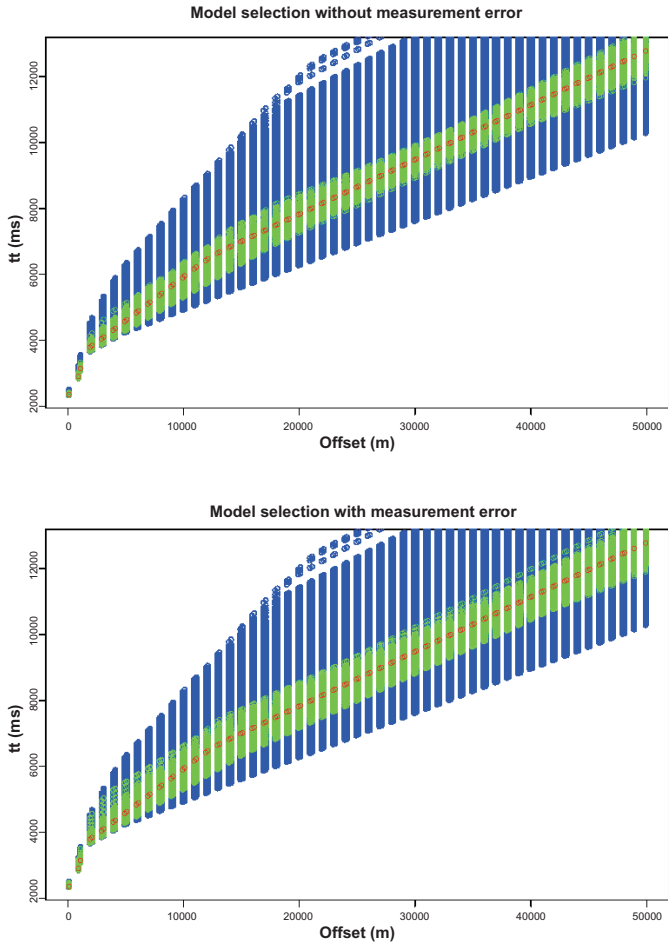

Figure 10: Rejection of implausible model space using the seismic emulator. The upper plot shows the result of assuming zero measurement error in the screening function, and the lower plot assumes a measurement error of 50 ms. The red dots show the result of running the model simulator for a chosen set of model parameters, the "target" travel time dataset. The emulator output is then compared for each of 10000 models which were not used to construct the emulator. For comparison, these 10000 models were then run through the full simulator and the outputs for those which were selected by the emulator are plotted using green dots and those which were rejected from running the emulator are plotted in blue. In the case where no measurement error is assumed, 213/10000 models were selected, and in the case where measurement error of 50ms is assumed, 498/10000 models were selected.

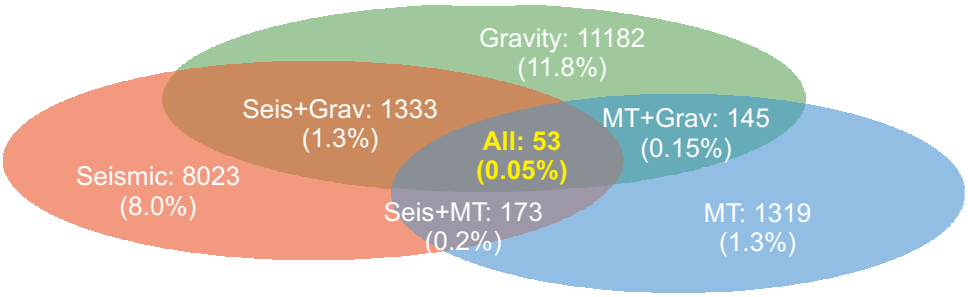

Figure 11: Model selection statistics using the seismic, gravity and MT emulators to jointly select plausible model space. Using these three emulators, it is seen that only 53/100000 (0.05%) of the model parameter space is plausible after a single cycle.

the methods, then the scaling factor γ_{seis}, γ_{MT} and γ_{grav} can be changed accordingly at the user's discretion. However, to ensure reliability of the system, each γ_x should be calibrated such that there is a high probability that, should for a synthetic example, the "true" set of model parameters be presented to the system, there is a very high probability that the model will be accepted. In our scenario here, we chose $\gamma_{seis} = \gamma_{MT} = \gamma_{grav} = 3$, which meant that 97% of the time, for a large number of target models, the "true" model parameter set (that used to generate the synthetic data) was deemed plausible.

2.9.1. Multi-cycle screening

Having used the emulators to test for plausibility and sampled the prior model space, we can then use this plausible region to construct new emulators, which, because they are built over a smaller model space, will generally have better predictive accuracy, and thus can be used to further constrain the plausible model space for a given target dataset. At each cycle, we therefore expect the uncertainty functions associated with each emulator to reduce in

magnitude. We ran the scheme over 11 cycles (after which there was no discernible reduction in the emulator uncertainty functions), at each cycle seeking to find 1000 plausible models with which to both build a new emulator, and reduce the size of the model space being screened. The number of candidate models generated at each cycle in order to find 1000 plausible ones is shown in Table 1. Because we are able to exclude a model as implausible on the basis that it is deemed implausible by any one of the seismic, spike, MT or gravity cases, we do not need to generate the emulator output for each method for every candidate model. However, for a candidate model to be deemed plausible and so be used in the subsequent cycle, it must "pass" the plausibility test in each case.

Note also that after the first cycle, the plausible model space to be searched isn't defined simply by the marginal parameter bounds, but also by the condition that a model was "passed" by the emulator screening from the previous cycle. Therefore in each cycle, for a candidate model to be deemed plausible, it must not simply be screened by the emulators generated using the plausible model space from the immediately previous cycle, but by those generated by all previous cycles. So, for a model to be deemed plausible on the 11th cycle, it must pass 44 screening tests (using each of the seismic, spike, MT, and gravity emulators from each of the previous cycles). However, if a model fails at any one of these stages, it is deemed implausible.

Because for a model to deemed implausible, only one "failure" is required, the computational efficiency of the screening process can be maximised by, for example, ordering the screening process such that the apparently most stringent screening method (seismic, spike, MT or gravity) is run first. This

can be tested using each screening method separately. In Figure 11, the MT selection is clearly the most stringent. However, in considering screening efficiency, it is also the case that each emulator (seismic, spike, MT or gravity) can take differing times to run (this is a function of the number of coefficients requiring computation in each case), as shown in Table 1. The efficiency of candidate model rejection is thus a function both of the emulator run time and the stringency of that emulator. In the example presented here, for each candidate model we screened using the gravity and spike emulators first, since these were the faster to run emulators (due to computing single data points rather than coefficients for data functions in the case of the seismic and MT cases).

On implementing this screening strategy, the marginal parameter histograms for the models deemed plausible are plotted in Figure 12. Taking a measure of the model space simply defined by a 16-dimensional box around the plausible parameter sets, the plausible model space volume has been reduced by a factor of 10^{-19} (or about 0.06 on average per model parameter). These histograms contain some information about the distribution of plausible models and in some instances be used to update the user's beliefs about the plausible parameter bounds. However, the marginal distributions often do not convey a large amount of the total information present in the distribution of plausible models. In particular, in the full joint distribution, there are likely to be inter-parameter relationships which emerge, for example, if the $v_2 \approx v_3$ then due to pseudo-non-uniqueness regarding the model specification, there may be a strong trade-off between s_2 and s_3. In this case the histograms would show wide distributions for s_2 and s_3 but this does

not imply that the system is at fault in failing to constrain these layers, but given the data uncertainty, and the fact that $v_2 \approx v_3$, the data do not constrain the thickness of these layers particularly tightly, whereas $s_2 + s_3$ may be constrained very effectively. Such dependencies are, however, indicative that it may be appropriate to reduce the number of model parameters since there is functional dependence between two or more of them.

The results presented here were obtained by generating candidate models using a marginal parameter sampling scheme; implementing a Sobol algorithm (Bratley and Fox, 1988) over a uniform distribution to select combinations of model parameters lying within the bounds. However, the screening process efficiency can be greatly increased by sampling from the joint distribution from the previous cycle, in that much less time is spent sampling redundant model space. We have tested some strategies for doing this with some success, though there are number of questions regarding the use of non-uniform prior parameter distributions at each cycle which need careful consideration, since the choice of prior distribution at each cycle influences the output parameter distributions: $P(X|A) = P(X).P(A|X)/P(A)$. Along with exploring appropriate parameterizations for a 3D structural scenarios, this is a subject of ongoing development.

3. Application to a real dataset

A dataset was kindly provided by Statoil, consisting of seismic, MT, and gravity data over a salt body. The free air gravity data are shown in Figure 14. We start by constructing emulators over the joint model space shown in Table 2. This model space was chosen after preliminary examination of

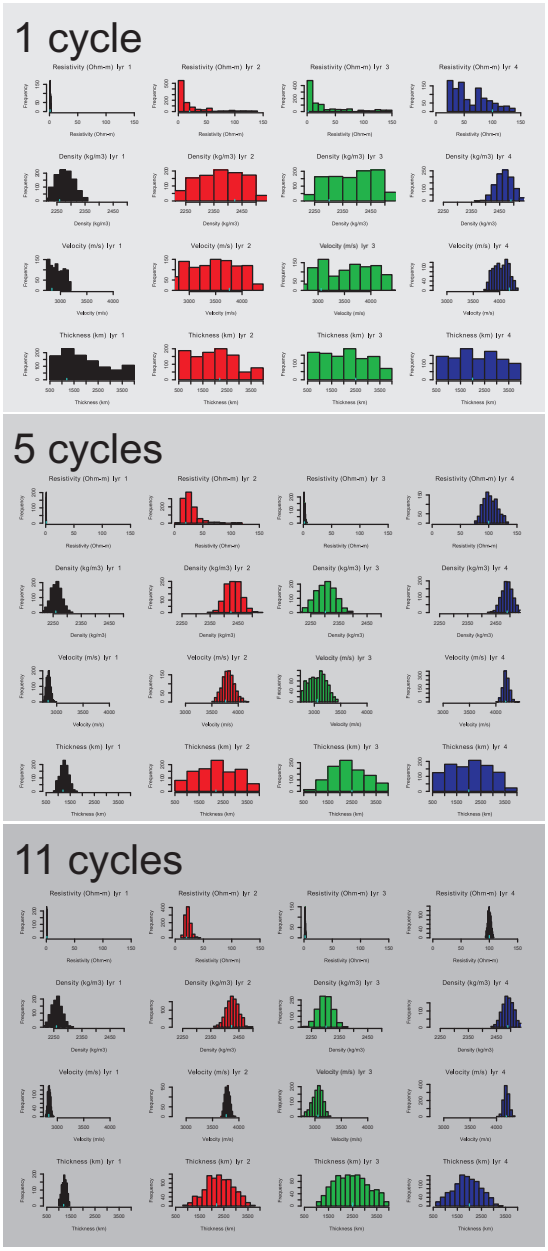

Figure 12: Histograms showing the marginal parameter distributions after 1 cycle (top), 5 cycles (middle) and 11 cycles (bottom). Notice that as further emulation cycles are carried out the plausible model space is reduced. Note that the widths of the distribution axes are the initial ranges defined by our prior model space (Figure 3).

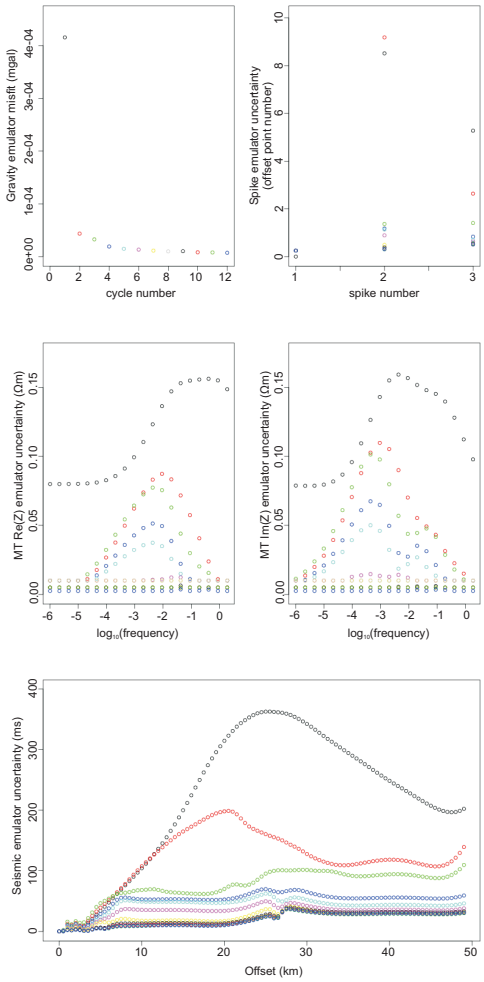

Figure 13: Misfit functions for each emulator. Bottom: seismic, Middle left: MT (real part), Middle right: MT (imaginary part), Top left: Gravity, Top right: spike. Misfit functions for cycles 1-11 are marked by the colours black, red, green, blue, cyan, magenta, yellow, grey, black, red, green and blue, respectively. Note that the uncertainty for successive emulation cycles decreases, reflecting the fact that each successive emulator for each technique is being built over a smaller model space and is thus able to capture more subtle data output variation than the emulator built in the previous cycle.

the data and some preliminary modelling to see which general class of models may be suitable. In order to test the methodology and avoid difficulties associated with designing a suitable conditional sampling strategy, at each emulation cycle we use a simple marginal sampling strategy, again sampling using a Sobol algorithm, from uniform distributions over the range of model parameters from the models deemed plausible from the previous cycle. Figures 15 and 16 show results from using only the *seismic* emulator to screen model parameter sets from a region over salt (MT7) and a region where there is no salt (MT12). The locations of MT7 and MT12 are shown in Figure 14. It can be seen from Figures 15 and 16 that the system is clearly detecting the salt body by virtue of the fact that layer 3 clearly has a salt-like velocity at MT7 whereas at MT12 this is not the case.

We then included the MT and gravity emulators in the screening process. To generate joint candidate models, we utilise other data provided by Statoil to elicit a relationship between resistivity and seismic velocity for the local region for both the salt and sedimentary regimes. The relationship for the sedimentary case is shown in Equation 34. As in the synthetic example, we use Gardner's relation for the relationship between seismic velocity and density. Using the information provided by Statoil, and because the relationship is empirical and uncertain, we also specify an uncertainty, this time normally distributed, on both the resistivity and density as a function of velocity (Equation 34).

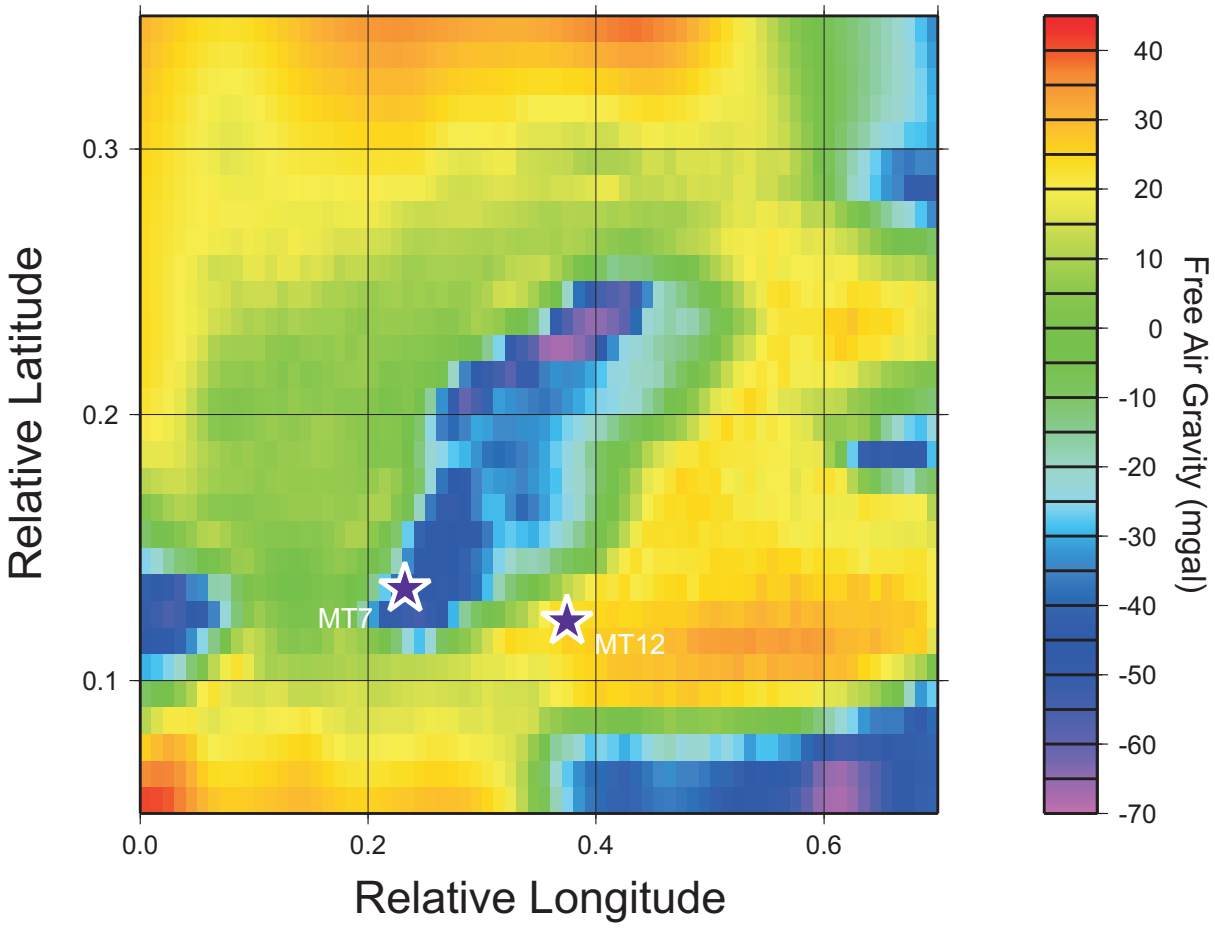

Figure 14: Gravity map in the region of MT7 and MT12. Note the presence of the salt body indicated by the low gravity anomaly under MT7.

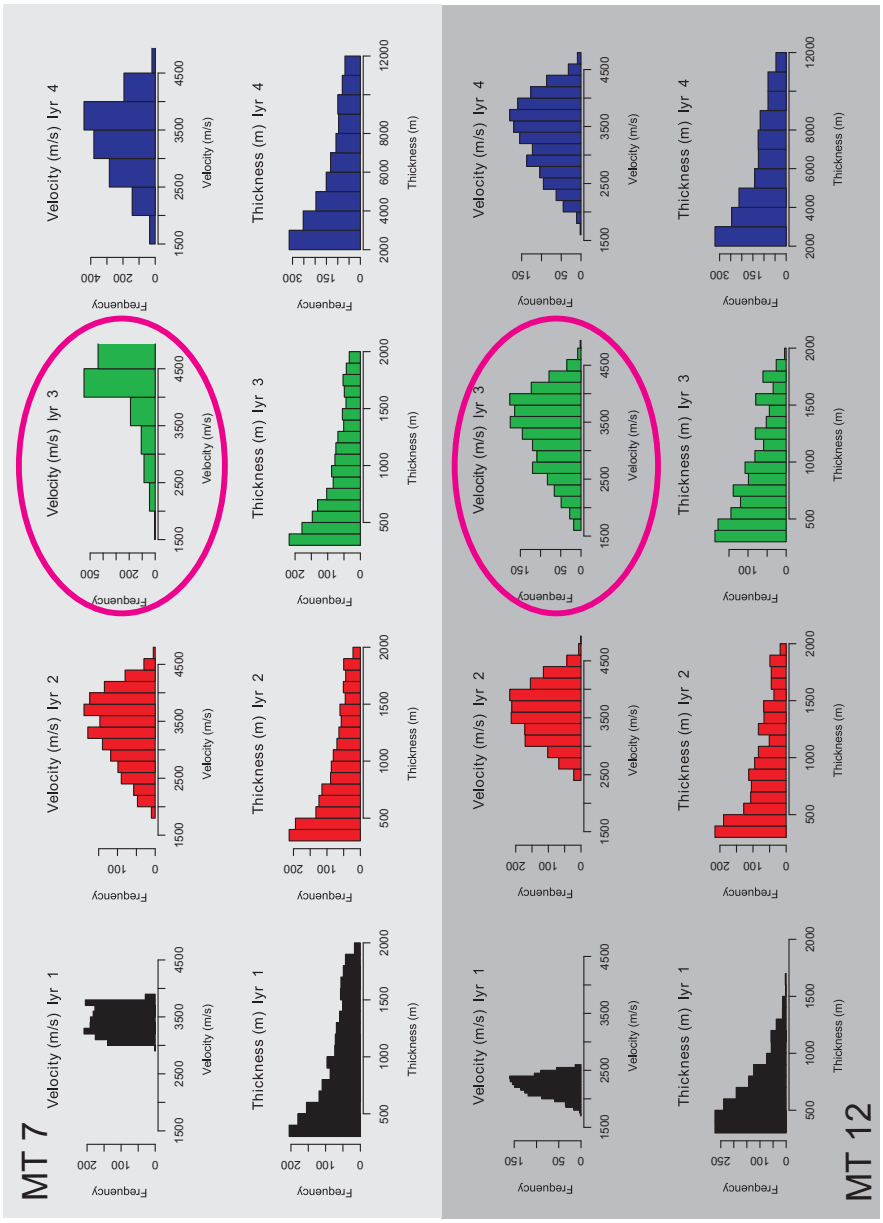

Figure 15: Histograms showing the marginal parameter distributions after 11 cycles screening only on seismic data for MT7 (left) and MT12 (right). MT7 is a profile over a salt body and MT12 is over sediment. Note that the salt structure is clearly being detected by the screening process in that a higher velocity in layer 3 (circled) is favored.

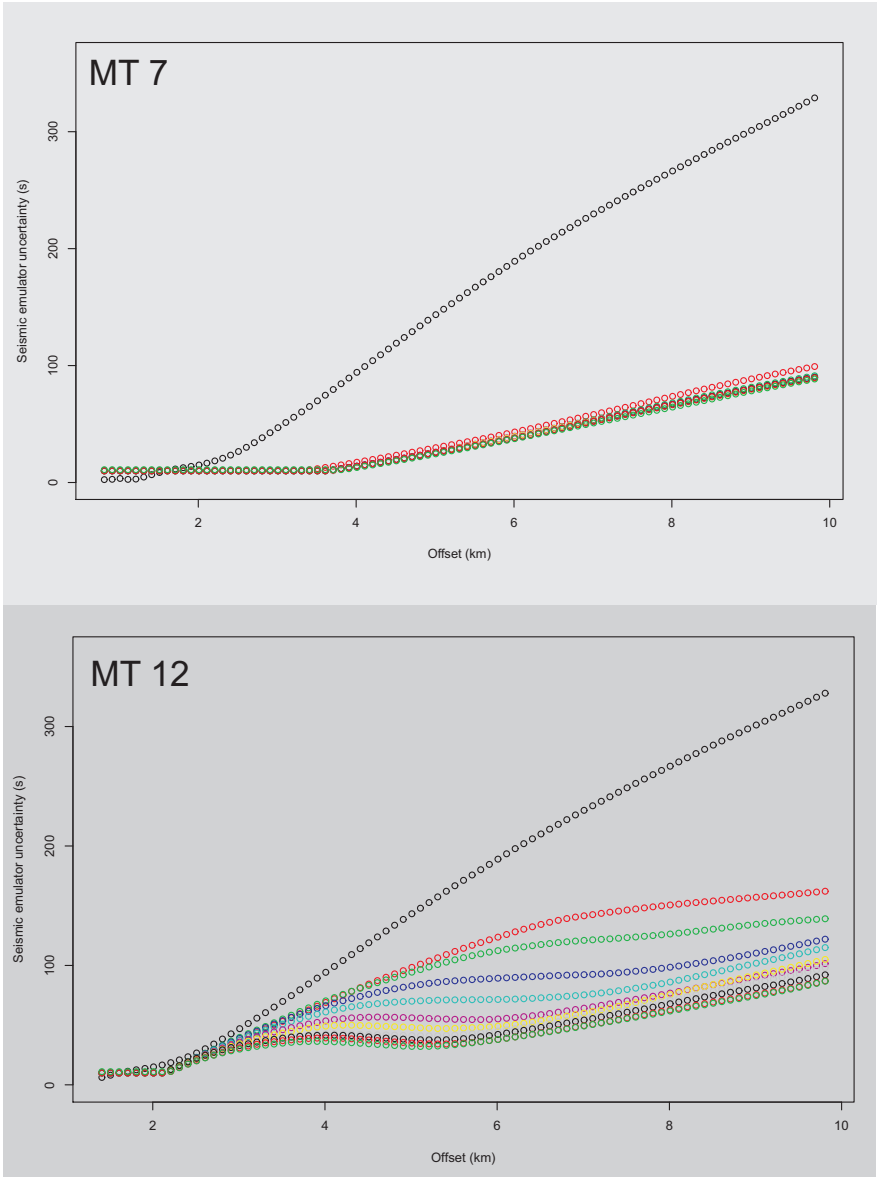

Figure 16: Misfit functions for each emulator screening only using the seismic emulator. Top: MT7 (over salt) Bottom: MT12 (over sediment).

$$\log(r) = 1.9421 + 2.4514 \times 10^{-5} v - 2.9587 \times 10^{-7} v^2 + 6.3142 \times 10^{-11} v^3 + N(0, 0.44) \tag{33}$$

$$\log(\rho) = 0.554 + 0.25 \log(v) + N(0, 0.0002) \tag{34}$$

From the same relationship data, the relationship for salt did not follow a discernible trend, but was seen to typically comprise velocities close to 4500 ms^{-1}, very high resistivity ($> 100\Omega\text{m}$) and density around 2160 ms^{-1}. After our initial analysis (Figure 16), we identified stations which appeared to lie over a salt body. In implementing the joint screening method, for these stations we specified that layer 3 should comprise salt.

On studying the system and observing that our method of model parameterization was causing a number of non-unique models to be generated as a result of the distribution of layer thickness parameters, we modified the sampling scheme to use gamma distributions (with shape parameter 3) in order to sample over the thickness parameter ranges.

Use of real field data necessitates the normalization of the gravity data to a particular total model thickness. This was not the case with the synthetic experiment because the "true" data were generated using the same simulator which was being used to generate the gravity emulator training datasets. The real data, however, were not generated with such a simulator and so the simulator output must be calibrated against the data. This was accomplished firstly by fixing the total model thickness to 12 km, and secondly by using one MT station (MT12) as a calibration station, in that we performed model screening simply using the MT and seismic datasets, then used the observed

relationship to give a distribution of plausible gravitational models at this station. The gravity simulator outputs for these models were then generated and the modal (most likely) gravity value from this output distribution used to calibrate an offset against the observed gravity value at this station. This calibration was then applied at each of the other stations, where gravity screening was included. Figure 17 shows the parameter histograms and uncertainty functions for MT7, located over a major salt feature. These were obtained using the seismic, gravity and MT emulators to screen the model space. Note that the spike emulator was not used here because the data did not contain sufficiently discernible gradient discontinuities for the spike emulator to be built reliably. Note, on comparing Figure 17 with Figure 15, we have also modified the prior velocity bounds (given by the range of the histogram axes in Figure 15). This was in order to ensure that joint models on the tail of the interparameter relationship distribution (Equation 34) were included in the distribution of candidate models.

4. Discussion

The synthetic example described in Section 2.9, demonstrates how the method is effective at screening model space for plausibility and how, in a multi-cycle regime, this emulator-based approach provides a means for quantifying the uncertainty associated with a modelling scenario.

In both the synthetic and real data examples, we see from the uncertainty plots in Figures 13 and 16 that the emulator uncertainty functions are reducing with increasing cycle. This shows that at each cycle, more structure is being obtained about the relationship between the model parameters

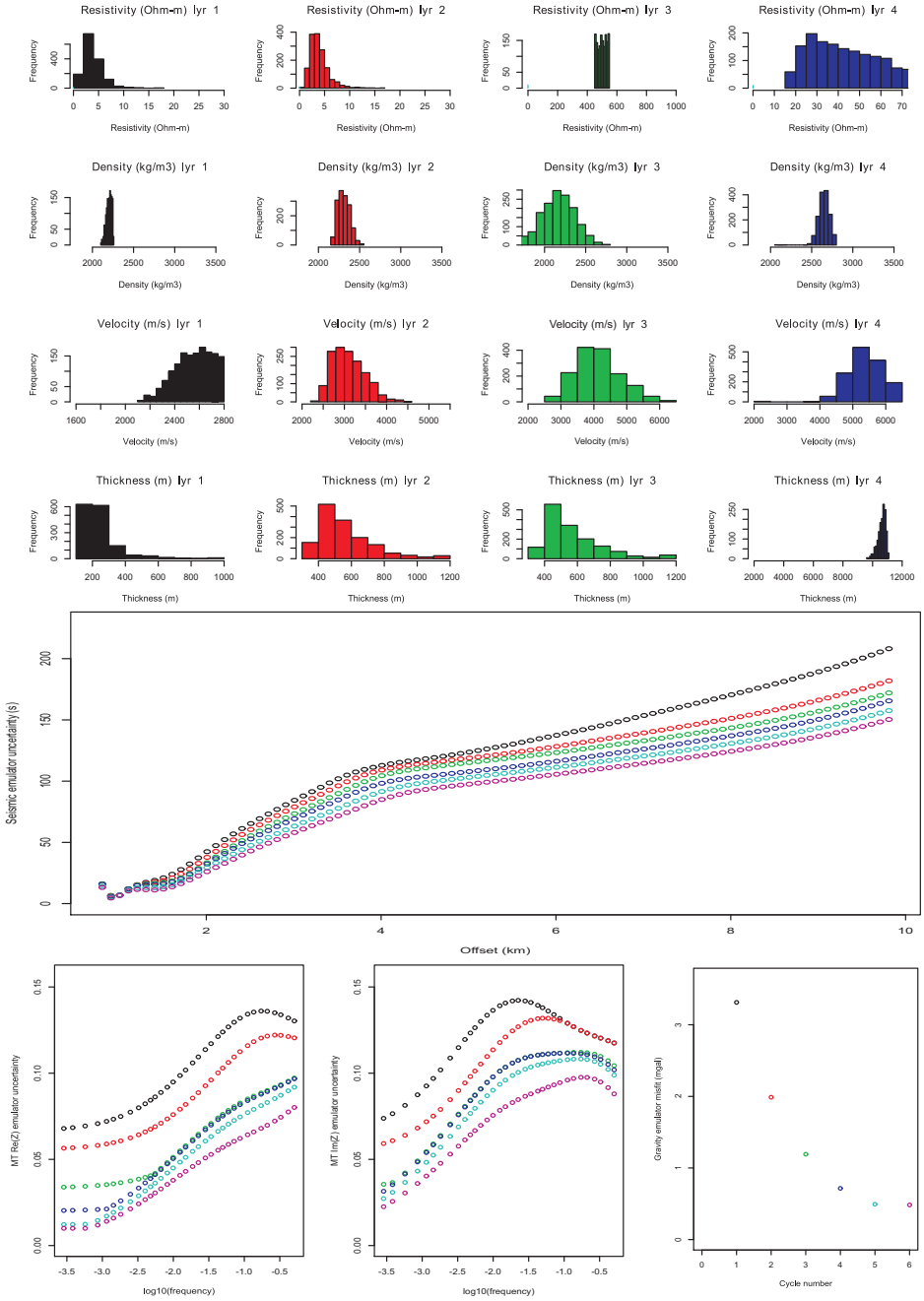

Figure 17: Histograms (top) showing plausible model parameters and emulator uncertainty functions (bottom) for a 1D profile at MT station MT7.

and the data. The limit of these uncertainty functions represents the total uncertainty associated with the system, given user specifications about the model space, the emulator parameterization, inter-parameter relationships, data uncertainty, model discrepancy and others.

Comparing Figures 13, 16 and 17 shows that in the case of the synthetic example the emulator uncertainty functions converge more efficiently than in the case of the real data. Primarily this is due to the fact that in the synthetic case, the class of model parameterization used to generate the synthetic dataset (4 layers) is of the same class as the candidate models being screened, whereas in the case of the real data, it is not the case that the earth structure which generated the data consists of four distinct layers. Hence, we also did not implement the spike emulator for screening in the case. This highlights a limitation in our current implementation and how by incorporating more user knowledge about the target structure, more informative results will be obtained through use of an intelligent and appropriate class of parameterization for the candidate models.

As discussed briefly in section 2.9.1, we have implemented a relatively simple marginal parameter sampling strategy. This means that model space is excluded conservatively and that the considerable amount of joint information contained in the full distribution of plausible models from the previous cycle is not used. In developing the methodology further, particularly in the case of more highly parameterized models, investigating conditional sampling strategies will doubtless increase the efficiency of the method further.

Our results show the value of a top-down Monte Carlo model screening approach, where the constraint process begins by considering the whole

model space, rather than seeking to iteratively search the model space for increasingly likely models. In particular, adopting such an approach removes the possibility of underestimating the uncertainty associated with the system through undersampling of the model space. The adoption of a simple Monte Carlo screening scheme also means that all kinds of uncertainty can be included in the analysis simply by generating the appropriate class of candidate models, and allows for straightforward and explicit specification of prior beliefs and uncertainties associated with the system, which is often not the case with a search-based deterministic approach.

4.1. Monte Carlo sampling or inverse solvers?

Although we are strongly advocating the use of a forward screening approach in order to fully sample prior model space, we are not proposing that this is the only valid "prescription" for geophysical constraint. The appropriate set of tools clearly depends on the system concerned, the degree of importance attached to understanding the uncertainty associated with the system, and the scale of the uncertainty in the problem relative to the size of the potential space within which plausible models are being sought. For systems where there is no, or very little, uncertainty present in the system, and thus where the plausible model space reduces to a point, or perhaps a collection of separated points, then a sampling method, such as that advocated here, will be very inefficient, since the probability of the system generating the one correct model will be very small (a 'needle in a haystack'). In this case, an inverse search-based solver will rapidly and usefully find an acceptable model. This result may then either be deemed sufficient in and of itself, or be used to update the scientist's understanding and beliefs about

the system, and allow a contraction of the relevant region of model space, over which a full sampling may then be carried out, perhaps facilitated by using an MCMC method.

Often, however, there is a considerable amount of uncertainty in the system, but in order to obtain a result this may be ignored and it is often the case that a single result is presented with a weakly substantiated statement regarding the uncertainty on the result. In this study we have presented a means by which a full sampling of the model space, which is required in order to make robust statements regarding the uncertainty, may be tackled.

4.2. Full simulator code or emulator?

For the 1D problem shown here, it would doubtless be more efficient to obtain a solution using the full simulator codes, and although we have used a simple scenario to demonstrate the method, we do not suggest that the emulation approach is necessarily the optimum way to obtain a solution for this particular problem.

For the purposes of screening model space, given that an emulator is trained using runs of the full simulator, it is a pertinent question as to whether it would simply be more efficient to use the full simulator outputs. In many ways the emulator described here can be thought of as interpolating the model space between 'known' points sampled by the full simulator by rapidly computing approximate outputs for the region between known points. For a very large model space and where the simulator is computationally expensive, and so the full simulator sampling is relatively sparse, it can greatly aid the ability to discern the extent of the plausible region to estimate the region between the sparsely known points. In cases where it is feasible to use the

full simulator code to sample the prior model space to a sufficient degree, then in deeming the sampling density sufficient, there is by definition no need to build an emulator. An emulator can therefore be considered useful whenever it would be helpful to be able to quickly interpolate between points of known model space. Once the plausible regions of model space have been clearly identified as plausible given the uncertainties in the system, then the user may wish to sample this region using the full simulator code.

As parallel computation methods are becoming increasingly commonplace, running a forward Monte Carlo screening scheme using the full simulators on a parallel system is clearly feasible. Indeed, the increasing use of parallel systems will doubtless make larger numbers of forward simulations feasible for larger problems too. However, given the size of the prior model spaces for many modern real-world geophysical problems, in order to screen the entirety of the model space, rather than adopting an MCMC or deterministic bottom-up search-based strategy, it is difficult to see how this could be achieved in the medium-term without a proxy-based method such as emulation, and based on the results presented here, and from examples in other fields, for example in Rougier et al. (2009) and Vernon and Goldstein (2009), we propose that emulation may provide a valuable tool for geophysical structural constraint.

5. Conclusions

In this paper we have shown that the method of emulation has the potential to make a large contribution in the field of geophysical modelling. From the results presented, it is clear that use of an emulation method makes pos-

sible the handling of entire model spaces, rather than small portions thereof. In the synthetic example, shown in Section 2.9.1, we have shown how, using a desktop computer workstation over a period of $\sim$72 hours, 245 million sets of model parameters could be screened for plausibility, and that in general use of an emulator affords a speed increase of several orders of magnitude in terms of the rate at which models can be screened for plausibility.

In applying the method to a real dataset, we have also shown that an emulator-based approach can be used to discern the plausible region of model parameter space for a practical problem. The approach taken here is quite simplistic in terms of the methodology and parameterization, however these results show that with the application of larger computational resources, emulators may make possible the handling very large model spaces for 3D systems throughout geophysics.

In the field of deep crustal imaging, it is commonly the case that uncertainties in data and in physics are not handled in a robust way, and prior beliefs or assertions about the system are often not explicitly stated. The result is that we often present an optimum model without a thorough assessment of the associated uncertainty. We have presented, using a simple geophysical example, an approach which seeks to practicably tackle the screening of the entire model space, with the aim of discerning all plausible regions, rather than adopting a search-based approach, which may be prone to not considering useful regions of model space, and to implicit conditioning through sometimes weakly justified choice of regularization and smoothing parameters.

Our aim here is not to present the model screening approach as the only

useful means for structural constraint. Indeed, as we discuss in Section 4, the inverse and search-based approaches are highly useful particularly when there is little uncertainty associated with the system and where there is strong information about the plausibility of the prior model space. We are, however, advocating that although they have not generally been employed in recent decades on the grounds of computational expense, model screening apporaches such as that presented here, are increasingly feasible, and that proxy-based emulators and other similar tools have the potential to help facilitate this kind of screening method.

6. Acknowledgements

The authors would like to thank ITF and the sponsors of the JIBA (Joint Inversion with Bayesian Analysis) consortium, through which this work was funded, for their financial provision and advice. The JIBA sponsors are: Statoil, Chevron, ExxonMobil, Nexen, Wintershall, RWE and Shell. We would also like to thank Statoil in particular for providing a dataset on which to test the method. Most of the coding for this study was carried out using R, an open source statistical coding environment, along with several additional modules (R Development Core Team, 2008; Dutang, 2009; Carnell, 2009).

7. Bibliography

References

Avdeev, D. B., Kuvshinov, A. V., Pankratov, O. V., Newman, G. A., 2002. Three-dimensional induction logging problems, part I: An integral equation solution and model comparisons. Geophysics 67, 413–426.

778 Bratley, P., Fox, B. L., 1988. Algorithm 659: Implementing sobol's quasi-
779 random sequence generator. ACM Transactions on Mathematical Software
780 14, 88–100.

781 Cagniard, L., 1953. Theory of the magneto–telluric method of geophysical
782 prospecting. Geophysics 18, 605–635.

783 Carnell, R., 2009. randtoolbox: Latin Hypercube Samples. R package version
784 0.5.

785 Dutang, C., 2009. randtoolbox: Generating and Testing Random Numbers.
786 R package version 1.07.

787 Gardner, G. H. F., Gardner, L. W., Gregory, A. R., 1974. Formation velocity
788 and density – the diagnostic basics for stratigraphic traps. Geophysics 39,
789 770–780.

790 Hastings, W. K., 1970. Monte Carlo sampling methods using Markov Chains
791 and their applications. Biometrika, 57, 97–109.

792 Heincke, B., Jegen, M., Chen, J., Hobbs, R. W., 2006. Joint Inversion of
793 MT, Gravity and Seismic Data appllied to sub-basalt Imaging. In: SEG
794 Technical Program Expanded Abstracts. Vol. 25. Society of Exploration
795 Geophysicists, pp. 784–789.

796 Hole, J. A., Ryberg, T., Fuis, G. S., Bleibinhaus, F., and Sharma,
797 A. K., 2006. Structure of the San Andreas fault zone at SAFOD from
798 a seismic refraction survey. Geophysical Research Letters, 33, L07312,
799 doi:10.1029/2005GL025194.

James, M. B., and M. H. Ritzwoller, 1999, Feasibility of truncated perturbation expansions to approximate rayleigh-wave eigenfrequencies and eigenfunctions in heterogeneous media: Bulletin of the Seismological Society of America, **89**, 433–441.

Jegen-Kulcsar, M., Hobbs, R. W., Tarits, P., Chave, A., 2009. Joint inversion of marine magnetotelluric and gravity data incorporating seismic constraints: preliminary results of sub-basalt imaging off the Faroe Shelf. Earth and Planetary Science Letters 282, 47–55.

Kennett, B. L. N., 2001. The Seismic Wavefield. Volume 1: Introduction and Theoretical Development. x+370 pp. Cambridge, New York, Melbourne: Cambridge University Press. ISBN 0 521 80945 2

Logemann, K., Backhaus, J. O., and Harms, I. H., 2004. A statistical emulator of the north-east Atlantic circulation. Ocean Modelling, 7, 97–110.

Lunnon, Z. C., Christie, P. A. F., White, R. S., 2003. An evaluation of peak and bubble tuning in sub–basalt seismology: modelling and results. First Break 21, 51–56.

Meier, U., A. Curtis, and J. Trampert, 2007. Global crustal thickness from neural network inversion of surface wave data: Geophys. J. Int., **169**, 706–722.

Moorkamp, M., Jegen, M., Roberts, A., Hobbs, R., 2011. A framework for 3-D joint inversion of MT, gravity and seismic refraction data. Geophysical Journal International 184(1): 477493

822 Podvin, P., Lecomte, I., 1991. Finite difference computation of traveltimes
823 in very contrasted velocity models: a massively parallel approach and its
824 associated tools. Geophysical Journal International 105, 271–284.

825 Press, F., 1970, Earth models obtained by Monte Carlo inversion: J. Geo-
826 phys. Res., **75**, 6575–6581.

827 R Development Core Team, 2008. R: A Language and Environment for Sta-
828 tistical Computing. R Foundation for Statistical Computing, Vienna, Aus-
829 tria, ISBN 3-900051-07-0.
830 URL `http://www.R-project.org`

831 Roberts, A. W., White, R. S., Christie, P. A. F., 2009. Imaging magmatic
832 rocks on the North Atlantic rifted continental margin. Geophysical Journal
833 International 179, 1024–1038.

834 Rougier, J., Sexton, D., Murphy, J. M., and Stainforth, D., 2009. Analyzing
835 the Climate Sensitivity of the HadSM3 Climate Model Using Ensembles
836 from Different but Related Experiments. J. Climate, 22, 3540–3557.

837 Sambridge, M., and K. Mosegaard, 2002, Monte Carlo methods in geophysi-
838 cal inverse problems: Reviews of Geophysics, **40(3),1009**, 1–24.

839 Shapiro, N. M., and M. H. Ritzwoller, 2002, Monte-carlo inversion for a global
840 shear-velocity model of the crust and upper mantle: Geophys. J. Int., **151**,
841 88–105.

842 Smith, A. F. M., and Roberts, G. O., 1993, Bayesian computation via the
843 Gibbs sampler and related Markov chain Monte-Carlo methods (with dis-
844 cussion). J. Roy. Stat. Soc. Series B 55, 3–23.

Vernon, I., Goldstein, M., 2009. Bayes Linear Analysis of Imprecision in Computer Models, with Application to Understanding Galaxy Formation. In: Augustin, T., Coolen, F. P. A., Moral, S., Troffaes, M. C. M. (Eds.), Proceedings of the Sixth International Symposium on Imprecise Probability: Theories and Applications. No. S061. Society for Imprecise Probability: Theories and Applications, pp. 441–450.

Zelt, C. A., and Barton, P. J., 1998. Three-dimensional seismic refraction tomography: a comparison of two methods applied to data from the Faeroe Basin. J. Geophys. Res., 103, 7187–7210.

Ziolkowski, A., Hanssen, P., Gatliff, R., Li, X., Jakubowicz, H., 2001. The use of low frequencies for sub–basalt imaging. In: 71st Annual International Meeting. SEG, pp. 74–77.

Cycle	Models tested
1	29,302
2	1,011,997
3	1,536,936
4	3,188,711
5	5,815,746
6	14,041,545
7	10,080,014
8	7,010,975
9	145,511,481
10	34,384,460
11	31,993,618
Total	245,532,785

Number of runs	Seismic simulator time	Seismic emulator time
10^4	45 minutes	15s
2.4×10^6	7.5 days	1 hour
5.76×10^7	180 days	1 day
Number of runs	MT simulator time	MT emulator time
10^4	12.8 minutes	1 minute
2.4×10^6	2.1 days	4 hours
5.76×10^7	50.4 days	4 days
Number of runs	Gravity simulator time	Gravity emulator time
10^4	3s	3s
2.4×10^6	12 minutes	12 minutes
5.76×10^7	4.8 hours	4.8 hours

Table 1: Left: Number of models tested in each emulator cycle (by each of the spike, gravity, MT and seismic emulators) in order to generate a population of 1000 plausible models. The histograms of the final selection are shown in Figure 12. Right: Typical run times for the simulator vs emulator for each of the seismic, MT and gravity cases on a high end workstation. The typical time required to construct the emulators ranged from 30 s in the case of the gravity emulator to about 10 minutes in the case of the seismic emulator. Note that the main factor controlling the emulator times is the number of α and β coefficients being used. As a result, while the MT simulator is considerably faster than the seismic simulator, the seismic emulator is faster to run than the MT emulator. In the case of the gravity emulator, the simulator is very simple and outputs only one point, and hence the emulator affords little advantage over the full simulator in this case. Comparison times for the spike emulator are not shown since it was run "piggy-backing" on the seismic emulator.

Layer:	1	2	3	4
Velocity (ms^{-1}):	1600-2800	2000-5500	2000-6500	2000-6500
Density (kgm^{-3}):	1800-3600	1800-3600	1800-3600	1800-3600
Resistivity (Ωm):	0.5-30	0.5-30	0.5-30	0.5-30
Thickness (m):	300-1000	300-1000	300-1000	300-1000

Table 2: Prior model space used for real dataset.